Due: 14 Feb 03

MA 3046 - Matrix Analysis

Programming Project #2

Least Squares and Adaptive Noise Cancellation

Consider the problem of trying to obtain a "clean" recording in a "noisy" environment. For example, suppose an after-dinner speaker is talking into a microphone while the guests are still finishing dessert, or we are trying to record a musician singing or playing in a noisy auditorium. We shall consider the speech or the music to be our desired "signal," i.e. what we really want to hear. (Although, for a number of real-world after-dinner speeches and rock musicians, precisely the opposite is probably the case.) In actual venues, however, the sound signal that the microphones will actually pick up is "contaminated" with an additive mix of the other room sounds, and will sound distinctly noisy.

But now suppose we also have N other, short-range microphones scattered around the room. We assume that these are located sufficiently far from the speaker that they do not pick up his or her voice, but that, between them, they do pick up essentially all of the background room noises. We further assume that this room noise is totally unrelated to whatever the speaker is doing or saying, or, in signal-processing parlance, that these background noises are *uncorrelated* with the speaker.

Now, let's start placing this problem in a mathematical framework. We let the desired signal, i.e. the speech or music that the lectern or stage microphone would have picked in the absence of noise be denoted by $s(t)$, and similarly, the signals picked up by the various room microphones be denoted by $n^{(i)}(t)\ ,\ \ i = 1, 2, \ldots, N$. Then, if the lectern in fact picks up a mix of the "signal" (speech or music), plus some combination of the room noises as also recorded by the room microphones, then we should expect that the total signal recorded by that microphone might have the form

$$m(t) = s(t) + \alpha_1 n^{(1)}(t) + \alpha_2 n^{(2)}(t) + \ldots + \alpha_N n^{(N)}(t)$$

Now, if we sample all of these signals, at times $t = t_1, t_2, \ldots, t_M$, to produce digitized versions of all of these signals, we would have the following room microphone signal (really noise) vectors

$$\mathbf{n}^{(i)} = \begin{bmatrix} n^{(i)}(t_1) \\ n^{(i)}(t_2) \\ \vdots \\ n^{(i)}(t_M) \end{bmatrix} \quad , \quad i = 1, 2, ..., N$$

a desired, "clean" speech signal vector of

$$\mathbf{s} = \begin{bmatrix} s(t_1) \\ s(t_2) \\ \vdots \\ s(t_M) \end{bmatrix}$$

and the lectern microphone signal (plus noise) vector of:

$$\mathbf{m} = \mathbf{s} + \alpha_1 \mathbf{n}^{(1)} + \alpha_2 \mathbf{n}^{(2)} + \cdots + \alpha_N \mathbf{n}^{(N)}$$

But recall earlier, we assumed that the various room noises were not correlated to the speech. In terms of the sample vectors we have here, this means that we expect that each of the noise vectors will be *orthogonal* to the speech vector, i.e.

$$\mathbf{s}^H \mathbf{n}^{(i)} = 0 \ , \quad i = 1, 2, \ldots, N$$

But if this is the case, then $\mathbf{s}$ should be orthogonal to the subspace spanned by the $\mathbf{n}^{(i)}$, and therefore our desired, noise-free signal should be nothing more than the component of the actual recorded lectern signal ($\mathbf{m}$) that is orthogonal to the subspace spanned by the $\mathbf{n}^{(i)}$!

But this should be an extremely simple algorithm to implement. We simply need to build any (least squares) orthogonal projection, say $\mathbf{P}$, onto the subspace spanned by the $\mathbf{n}^{(i)}$, then apply $\mathbf{I} - \mathbf{P}$ to the actual recorded lectern microphone signal ($\mathbf{m}$), and we should recover our "clean," noise-free signal.

Assignment:

Attempt to implement this algorithm for the noisy lectern microphone signal contained in the file **lectern_mike.wav**, and the room (noise-cancelling) microphone signals contained in the files **room_mike1.wav** through **room_mike6.wav**. (All of these are downloadable from the course web page.) Specifically, write a MATLAB **m**-file which will:

a. Use the MATLAB command **wavread** to convert each of these into a sample vector. (You may wish to confirm you've done this correctly by playing back these vectors using the MATLAB commands **audioplayer** and **play**. If you do, be sure to use a sample rate of 8000Hz.)

b. Using noise vectors your program from part a., and any MATLAB routines you wish, find, in a reasonably efficient manner, any matrix $\mathbf{P}$, that represents a projection onto the subspace spanned by the noise mike vectors.

c. Using the matrix **P** found in part b., find the projection of the lectern microphone signal vector onto the orthogonal complement of the subspace spanned by the noise vectors. This should represent the "clean" speech signal.

d. Using the MATLAB commands **audioplayer** and **play**, and a sampling rate of 8000Hz, play back the clean speech signal found in part c.

e. Briefly describe the MATLAB command(s) you used to accomplish part c. above. For each, indicate why you chose that particular command, and how the results from that command were used. Analyze the degree to which any behavior observed in parts a-d. above either does or does not agree with the theory discussed in class and in this handout.

References:

[1] Carlos Borges, private correspondence providing the basis and notation for this example.